Explorations in Learning to Code Through Games

Kara James
CU Boulder, Computer Science Undergrad
kara.james@colorado.edu

Gabriel Lennon
CU Boulder, Computer Science Undergrad
gabriel.lennon@colorado.edu

Emily Owens
CU Boulder, Computer Science Undergrad
emily.f.owens@colorado.edu

Joshua Weaver
CU Boulder, Computer Science Undergrad
joshua.weaver@colorado.edu

ABSTRACT
This paper discusses the process used to design a prototype for an educational coding game. It will discuss the requirements outlined for this project. It will then discuss the design of our final prototype, while also evaluating the user feedback we have received on our various intermediate designs.

INTRODUCTION
The idea of this project was to create an educational game that would introduce users to basic computer science control structures. This consists of two goals. The first goal is to make a system that provides people with applicable knowledge, presented in a way that encourages understanding and remembering the material. The second goal is to have that system also be entertaining to use, so the user will enjoy their learning experience.

REQUIREMENTS
Our project aims to teach the user fundamental programming concepts within the context of a video game. One of the key challenges we knew we needed to address in this project was how to make the game both instructive and accessible to anyone without a serious computer science background. Our project is not intended for use by those already versed in control structures and data types, but rather for those who are interested in coding, yet are intimidated by its somewhat abstract nature. Our system would guide the user through writing various functions that would control the movement of an onscreen player character. The user would navigate this character through levels that require different approaches, encountering new control structures and concepts as the difficulty of the levels increased. Given that our design included a difficulty curve, we wanted to ensure that the user received useful feedback from the game upon failing a level, so as to avoid unnecessary frustration. Since the concept of writing functions to control an onscreen object is already rather "computer science-y", one of our biggest challenges was developing an interface that made sense to users without any previous computer science exposure, and creating systems to display this information in a straightforward way. There are a number of systems available at this time that provide a similar approach to teaching programming, most of which can be found in the citations at the end of the paper[1].

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

DESIGN
Our final design consists of two modes: a normal mode and a challenge mode. The normal mode is basically what our initial design proposal was; the player writes functions, such as move left or jump, and maps those to keys on their keyboard. They then enter the play screen, take control of a character and attempt to complete levels using the actions they programmed. The levels become progressively more difficult, which requires the player to build additional functions or alter their existing functions. The challenge mode closely resembles one of our divergent designs. Here, instead of writing the functions, the player is provided with existing functions (possibly the functions that they wrote in normal mode). They use these functions to write a script for a robot to execute. When the player presses play they will observe the robot performing the sequence of actions the player designed for it. The player's objective is to get the robot to the end of the level without dying or getting stuck. Both modes in our design have a short tutorial that instructs the player on how the game works. Each mode also has a game over screen that will give hints on how to fix their mistakes and allow the user to return to the appropriate function building screen.

EVALUATION
The final design was largely successful in dealing with our identified challenges. Users found the majority of the interface to be clear and accessible, with few exceptions.

Our final design includes a number of dialogue boxes that include basic instructions for the complicated function building screens which follow. After we added the dialogue boxes to the design, fewer users had issues understanding how to navigate both of the function building screens. The only complaint we received on the final design was to include a clearer method of dismissing the dialogue box. They appreciated the dialogue boxes, and mentioned them as a particularly helpful element of our design.

However, the dialogue boxes didn't solve every issue in our design. In particular, the limited color palette we used in our design caused some confusion. Buttons had a unique color that we believed would differentiate them from all other elements on the screen, but a small number of users consistently had trouble telling whether or not particular elements were buttons. It will require further testing to determine an ideal way of distinguishing buttons, but a more expanded color scheme would likely be helpful.

These were the only major issues that users found. Users were able to properly select which Game Mode they wanted to play, and were able to change Game Options at will. They also appreciated the ability to view the level beforehand, as that allowed them to easily tailor their functions specifically for the challenges ahead.

This brings us to the other major challenge we faced: how to make the game instructive. This was a much more difficult thing to test, because we did not have a working function building screen to present to the users. We explained the basic theory of the screens to them, and they seemed to understand. The limitations in our prototype means that they were unable to see how their code actually affected their character's movement. Our users weren't able to provide large volumes of feedback on elements they only verbally walked through, however the feedback we did get was mostly promising.

Since our design has two different Game Modes, we decided to focus on different elements of coding in each. The normal mode is designed to teach users the basic elements that make up a single function, and then force the user to apply this knowledge several different times in order to assign different buttons to different actions. In theory, this should cause the users to gain a deeper understanding of what each line of code does. This is backed by the fact that some users were able to more accurately describe how a function would be built the second time they were interviewed.

The Challenge Mode is designed to teach users how a program executes, and users seemed to understand this with a small amount of prompting. The "Controls", "Variables", and "Functions" buttons were initially confusing, but most users were able to gain a rudimentary understanding of their purpose. Again, it seems likely that repetition will improve their understanding of these concepts.

CONCLUSION

This limited scope of this project did not allow us to preform adequate user testing on the educational techniques employed in the game. However, our design evaluators provided us with positive feedback, and seemed to understand the techniques employed in the prototype. This suggests to us that this design was very successful, and would be well worth developing.

ACKNOWLEDGMENTS

We would like to thank our professor, Dr. Clayton Lewis, for his continued support throughout this project. We would also like to thank all of our design evaluators, who provided invaluable feedback on our design iterations.

REFERENCES

1. Games that Teach Programming: A Brief Overview. http://www.gamasutra.com/blogs/RobLockhart/20130905/199667/Games_That_Teach_Programming_A_Brief_Overview.php.